Тестовые данные

Таблица 1 Таблица 2

|  |  |  |  |
| --- | --- | --- | --- |
| **city** | **type** | **cost** | **reportdate** |
| null | Peach | 632 | 31.03.2020 |
| Paris | Peach | 268 | 01.04.2020 |
| Paris | Peach | 115 | 02.04.2020 |
| Paris | Apple | 303 | 31.03.2020 |
| Paris | Apple | 4 | 01.04.2020 |
| Paris | Apple | 734 | 02.04.2020 |
| Paris | Pinapple | null | 31.03.2020 |
| Paris | Pinapple | 600 | 02.04.2020 |
| Paris | Pinapple | 160 | 03.04.2020 |
| Lyon | Peach | 337 | 31.03.2020 |
| Lyon | Peach | 740 | null |
| Lyon | Peach | 62 | 02.04.2020 |
| Lyon | Apple | 983 | 31.03.2020 |
| Lyon | Apple | 509 | 01.04.2020 |
| Lyon | Apple | 609 | 02.04.2020 |
| Marseille | Watermelon | 869 | 01.04.2020 |
| Marseille | Watermelon | 129 | 04.04.2020 |
| Marseille | Watermelon | 761 | 05.04.2020 |

|  |  |  |  |
| --- | --- | --- | --- |
| **city** | **type** | **cost** | **reportdate** |
| Paris | Peach | 690 | 31.03.2020 |
| Paris | Peach | 95 | 01.04.2020 |
| Paris | Peach | 50 | 02.04.2020 |
| Paris | Apple | 538 | 31.03.2020 |
| Paris | Apple | 303 | 01.04.2020 |
| Paris | Apple | 836 | 02.04.2020 |
| Paris | Pinapple | 137 | 31.03.2020 |
| Paris | Pinapple | 849 | 02.04.2020 |
| Paris | Pinapple | 900 | 03.04.2020 |
| Lyon | Peach | 20 | 31.03.2020 |
| Lyon | Peach | 255 | 01.04.2020 |
| Lyon | Peach | 706 | 02.04.2020 |
| Lyon | Apple | 284 | 31.03.2020 |
| Lyon | Apple | 958 | 01.04.2020 |
| Lyon | Melon | 790 | 02.04.2020 |
| Lyon | Pinapple | 535 | 01.04.2020 |
| Lyon | Pinapple | 29 | 04.04.2020 |
| Lyon | Pinapple | 610 | 06.04.2020 |
| null | Apple | 438 | 06.04.2020 |

Задания

Напишите SQL-запросы, которые выполнят следующее:

**Ответ**

Допустим, что:

**table1** – таблица 1

**table2** – таблица 2

1. Выведите всю информацию по городу Paris из таблицы 1

SELECT \* FROM table1

WHERE city = "Paris";

1. Выведите все уникальные даты из таблицы #city\_from\_base1, где есть информация по городу Lyon для ананасов (Pinapple)

SELECT DISTINCT reportdate FROM table1

WHERE city = 'Lyon' and type = 'Pinapple';

1. Выведите сумму всех стоимостей в одном запросе по датам и по городам и по типам товара из таблицы 2

SELECT reportdate, city, type, SUM(cost) from table2

GROUP BY reportdate, city, type

1. Покажите все строчки где есть null из таблицы 2

Если null – строчка:

SELECT \* FROM table2

WHERE city = 'null'

OR cost = 'null'

OR type = 'null'

OR reportdate = 'null'

Если null – тип:

SELECT \* FROM table2

WHERE city IS NULL

OR cost IS NULL

OR type IS NULL

OR reportdate IS NULL

1. Выведите уникальный список городов и товаров из обеих таблиц

SELECT city from table1 UNION SELECT city from table2;

SELECT type from table1 UNION SELECT type from table2;

1. Найдите такие типы товаров для Lyon, которые отсутствуют в таблице 2, но которые есть в таблице 1

SELECT DISTINCT table1.type from table1

WHERE table1.city = 'Lyon'

AND table1.type NOT IN (SELECT table2.type FROM table2

WHERE table2.city = 'Lyon');

1. Найдите все даты из таблицы 1, которых нет в таблице 2

SELECT DISTINCT table1.reportdate AS t1\_date from table1

where t1\_date NOT IN (SELECT table2.reportdate from table2)

1. Просуммируйте стоимость в таблицах для города Paris, по типам товара

SELECT type, SUM(cost) FROM

(SELECT \* from table1

UNION

SELECT \* FROM table2)

WHERE city = 'Paris'

GROUP BY type

1. Объедините данные из двух таблиц, найдите максимальную и минимальную стоимость по каждому типу товаров и в каждом городе на каждую дату. Выведите в виде одной таблицы формата – город, дата, тип товара, макс. стоимость, мин. Стоимость

SELECT city, reportdate, type, MAX(cost), MIN(cost) FROM

(SELECT \* from table1

UNION

SELECT \* FROM table2)

GROUP BY type, city, reportdate

ORDER BY city

1. На основе предыдущего задания, добавьте к каждой строке ближайшую предыдущую дату, (если имеется) и добавьте в вывод предыдущую дату, макс.стоимость и мин.стоимость на эту дату. (можно использовать cte, временные таблицы и любые альтернативные подходы)

*Пояснение:*

Задача решена на SQLite

**temp\_table** – просто объединенная таблица1 и таблица2

**group\_temp\_table** – сгруппированная таблица по городу, типу и дате из задания 9

**date\_str** (колонка) – месяц + день как строка без точек, например 21.09.2000 будет «0921». Это сделано, чтобы можно было сравнивать даты, т.к. SQLite не поддерживает тип дат и их нельзя просто так сравнить

**nearest\_date** (колонка) **–** ближайшая дата

**nearest\_max\_cost** (колонка) –максимальная цена на эту дату

**nearest\_min\_cost** (колонка) –минимальная цена на эту дату

DROP TABLE IF EXISTS temp\_table;

DROP TABLE IF EXISTS group\_temp\_table;

CREATE TEMP TABLE temp\_table AS

SELECT \* from table1

UNION

SELECT \* FROM table2;

CREATE TEMP TABLE group\_temp\_table AS

SELECT city, reportdate, type, MAX(cost) AS max\_cost, MIN(cost) AS min\_cost

FROM temp\_table

GROUP BY type, city, reportdate

ORDER BY city;

ALTER TABLE temp\_table ADD COLUMN date\_str;

UPDATE temp\_table SET date\_str = SUBSTRING(temp\_table.reportdate, 4, 2) || SUBSTRING(temp\_table.reportdate, 1, 2);

ALTER TABLE group\_temp\_table ADD COLUMN date\_str;

UPDATE group\_temp\_table SET date\_str = SUBSTRING(group\_temp\_table.reportdate, 4, 2) ||

SUBSTRING(group\_temp\_table.reportdate, 1, 2);

ALTER TABLE group\_temp\_table ADD COLUMN nearest\_date;

UPDATE group\_temp\_table SET nearest\_date = (SELECT t.reportdate FROM temp\_table t

WHERE t.date\_str = (SELECT MAX(t.date\_str) AS nearest FROM temp\_table t

WHERE t.date\_str < group\_temp\_table.date\_str));

SELECT DISTINCT gt.city, gt.reportdate,

gt.type, gt.max\_cost,

gt.min\_cost, gt.nearest\_date,

(SELECT MAX(cost) FROM temp\_table t

WHERE cost != 'null'

GROUP BY t.reportdate

HAVING gt.nearest\_date = t.reportdate) AS nearest\_max\_cost,

(SELECT MIN(cost) FROM temp\_table t

WHERE cost != 'null'

GROUP BY t.reportdate

HAVING gt.nearest\_date = t.reportdate) AS nearest\_min\_cost

FROM group\_temp\_table AS gt;

DROP TABLE IF EXISTS temp\_table;

DROP TABLE IF EXISTS group\_temp\_table;

Если у вас есть несколько способов решения, можете присылать все возможные варианты.

Сайт где можно выполнить запросы - <https://sqliteonline.com/>